

# **EXHIBIT F**

**FILED UNDER SEAL**

1 CLEMENT SETH ROBERTS (STATE BAR NO. 209203)  
croberts@orrick.com  
2 BAS DE BLANK (STATE BAR NO. 191487)  
basdeblank@orrick.com  
3 ALYSSA CARIDIS (STATE BAR NO. 260103)  
acaridis@orrick.com  
4 EVAN D. BREWER (STATE BAR NO. 304411)  
ebrewer@orrick.com  
5 ORRICK, HERRINGTON & SUTCLIFFE LLP  
The Orrick Building  
6 405 Howard Street  
San Francisco, CA 94105-2669  
7 Telephone: +1 415 773 5700  
Facsimile: +1 415 773 5759  
8  
9 SEAN M. SULLIVAN (*pro hac vice*)  
sullivan@ls3ip.com  
10 MICHAEL P. BOYEA (*pro hac vice*)  
boyea@ls3ip.com  
11 COLE B. RICHTER (*pro hac vice*)  
richter@ls3ip.com  
12 LEE SULLIVAN SHEA & SMITH LLP  
656 W Randolph St., Floor 5W  
Chicago, IL 60661  
13 Telephone: +1 312 754 0002  
Facsimile: +1 312 754 0003  
14  
15 *Attorneys for Sonos, Inc.*  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

UNITED STATES DISTRICT COURT  
NORTHERN DISTRICT OF CALIFORNIA  
SAN FRANCISCO DIVISION

GOOGLE LLC,  
Plaintiff and Counter-defendant,  
v.  
SONOS, INC.,  
Defendant and Counter-claimant.

Case No. 3:20-cv-06754-WHA  
Related to Case No. 3:21-cv-07559-WHA

**REBUTTAL EXPERT REPORT OF  
DOUGLAS C. SCHMIDT**

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 and Lambourne on October 20, 2011 detailing staffing requirements for “Play-to-Sonos”  
 2 initiative

3 • SONOS-SVG2-00026246 – email from Mr. Coburn sent November 2, 2011 to Messrs.  
 4 Lambourne, Kuper, Schulert, and Millington and Ms. Hoadley scheduling meeting to  
 5 discuss “Play-to-Sonos” initiative

6 • SONOS-SVG2-00026246 – email from Mr. Coburn sent to Messrs. Lambourne, Kuper,  
 7 Schulert, and Millington and Ms. Hoadley on November 7, 2011 summarizing action  
 8 plan from meeting regarding the “Play-to-Sonos” initiative

9 • U.S. Patent Appl. No. 13/341,237 – patent application describing the “Play-to-Sonos”  
 10 initiative filed on December 30, 2011, which I understand took several weeks (at a  
 11 minimum) to prepare<sup>15</sup> and, given the holidays, was therefore likely started in November  
 12 2011.

**IX. OVERVIEW OF THE PRIOR ART**

13 153. In this section, I provide a respective overview of the primary and secondary  
 14 references that Dr. Bhattacharjee relies upon.

**A. YouTube Remote (YTR) System****1. General Overview of YTR System**

154. Dr. Bhattacharjee provides opinions that the Asserted Claims of the '033 Patent (i.e.,  
 16 claims 1-2, 4, 9, 11-13, and 16) and Asserted Claims 14-15, 18-19, and 25 of the '615 Patent are  
 17 anticipated or rendered obvious by a YouTube Remote (“YTR”) System.

18 155. As explained in more detail below, the YouTube Remote (“YTR”) System included:  
 19 (1) a YTR software application installed on one or more computer devices, such mobile phones,  
 20 that were also referred to as “remotes”; (2) a “Lounge Server,” also referred to as an “MDx Server,”  
 21 that operated using an “MDx Protocol”; and (3) one or more “Leanback Screens,” such as TVs,  
 22 that were also referred to as “Connected Screens” (or simply “Screens” for short). According to  
 23 Dr. Bhattacharjee, “[t]he YTR application could be used for local playback on the mobile device  
 24 (‘Local Playback Mode’),” but “[w]hen video playback was transferred from the YTR application  
 25 on the mobile device to one or more Screens, the YTR application would serve as a remote control  
 Report, ¶157.

156. Dr. Bhattacharjee points to three versions – Versions 1-3 – of the YTR application

---

28 <sup>15</sup> This timeline is consistent with my own patent-filing experience. See also, e.g.,  
 29 <https://milleripl.com/blogs/patents/how-long-does-it-take-to-get-a-patent> [SONOS-SVG2-  
 30 00226859].

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 Leanback Screens to the Lounge Server until Version 3 of the MDx Protocol. *See, e.g.*, Levai Dep.  
2 Tr. (May 4, 2022), 107:3-9, 112:7-114:10; Levai Dep. Tr. (January 6, 2023), 34:4-14. This means  
3 that the first time the MDx Protocol relied on a “remote playback queue” instead of a “local  
4 playback queue” was not until after January of 2014. *Id.*

5 160. According to a YouTube press release, a “beta” version of the YTR application for  
6 mobile phones running the Android operating system appears to have been released on November  
7 9, 2010. GOOG-SONOS-WDTX-INV-00015413, 13. As explained in the title of the press release,  
8 the YTR application could be used to “[c]ontrol YouTube on the desktop, or the TV.” *Id.* To  
9 enable this functionality, the YTR application “create[d] a virtual connection between your phone  
10 and YouTube Leanback” on a “Google TV or computer.” *Id.* The connected TV or desktop device  
11 was referred to as a “Leanback Screen.”

12 161. As explained in the press release: “To ‘pair’ your phone with your Leanback screen,  
13 simply sign into YouTube Remote on your Android phone, and to YouTube Leanback on your  
14 Google TV or computer with the same YouTube account. Just like that, you’ve connected your  
15 powerful multi-touch Android screen with the biggest screen in your home.” *Id.* As Dr.  
16 Bhattacharjee has explained, “[a] YTR application and one or more [Leanback] Screens are paired  
17 together by logging into the same YouTube account which registers them with the Lounge Server  
18 and adds them to a lounge ‘session.’” Bhatta. Op. Report, ¶178; *see also id.*, ¶204 *citing* ’998  
19 Patent, 4:21-32 (“For example, pairing can involve having the remote control and control[led]  
20 devices login to the same ‘user account’ and then notifying the server that they are ‘connected to  
21 the network’ to initial a ‘session.’”).

22 162. I understand that the pairing of the YTR application with a Leanback Screen was  
23 facilitated by an intermediary YouTube cloud server referred to by Google as the “Lounge Server”  
24 or “MDx Server.” *See* Bhatta. Op. Report, ¶158.

25 163. After a “virtual connection” (or “pairing”) was established, I understand that a user  
26 could then use the YTR application to control playback of videos on the Leanback Screen. GOOG-  
27 SONOS-WDTX-INV-00015413, 13-14. To the extent the YTR application could be paired with  
28 multiple Leanback Screens in a session, it is my understanding that the same media would be played

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

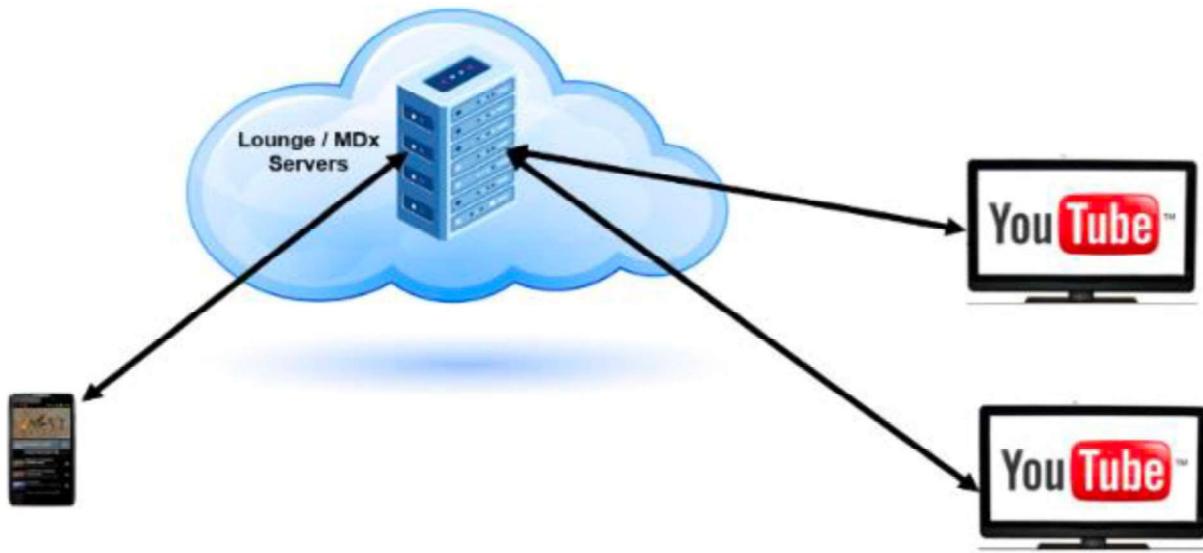
1 on ***all*** the Leanback Screens and that the YTR application and/or cloud-based Lounge Server would  
2 be configured to control the playback on ***all*** paired Leanback Screens ***collectively*** (as opposed to  
3 controlling any one of the multiple Leanback Screens ***individually*** or controlling a subset of the  
4 multiple Leanback Screens). *See* Bhatta. Op. Report, ¶¶158, 178-79, 183; GOOG-SONOS-  
5 WDTX-INV-00015423, 24. I understand that it was possible to add a new Leanback Screen to an  
6 already existing session involving a different Leanback Screen, as long as the Leanback Screens  
7 were logged into the same YouTube account. Levai Dep. Tr. (January 6, 2023), 45:23-46:5.

8 164. I also understand that the YTR application could be used on more than one computer  
9 device (e.g., mobile phone) in the same session. Levai Dep. Tr. (May 4, 2022), 105:15-106:19;  
10 Levai Dep. Tr. (January 6, 2023), 39:5-12, 43:14-45:22. In other words, multiple Leanback Screens  
11 could be controlled by multiple YTR remote controls in the same session. Bhatta. Op. Report, ¶179  
12 (“A lounge session may have multiple devices associated with it, in particular a single lounge  
13 session may support multiple Leanback Screens (LoungeSession.java23, lines 46, 204). *See also*  
14 LoungeSession.java, lines 21-23 (‘A lounge session consists of several devices (remote controls  
15 and players) that communicate through the remote control server.’)”). I further understand that a  
16 second remote control could be added to an existing session between a first remote control and one  
17 or more Leanback Screens, and that the second remote control, like the first remote control, would  
18 obtain a copy of the entire list of videos for playback. Levai Dep. Tr. (January 6, 2023), 43:14-  
19 45:22.

20 165. According to Dr. Bhattacharjee, each YTR application/remote control allowed a  
21 user to not only “create queues of videos,” but also to “edit the queue by, for instance, removing  
22 items from the queue or rearranging their order.” Bhatta. Op. Report, ¶¶165-66. Since multiple  
23 YTR remote controls could be used in a session, multiple users could create and edit playback  
24 queues on multiple Leanback Screens.

25 166. As discussed in Video #2 cited by Dr. Bhattacharjee, the connection between a YTR  
26 application and a Leanback Screen could be established when a mobile phone installed with the  
27 YTR application was connected to a local area network (LAN), such as a home Wi-Fi network, or  
28 when it was only connected to a wide area network (WAN), such as a 3G cellular network. *See*

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY



170. It is also important to note that only “videoIDs” are maintained in the playback  
 12 queues on the remote controls and Leanback Screens. Bhatta. Op. Report, ¶¶193-97; GOOG-  
 13 SONOSWDTX-00040194; Levai Dep. Tr. (May 4, 2022), 46:25-54:9;  
 14 RealPlaystatePushService.java, lns. 41-350 (57, 137-47) (bigScreenPlaylist); LeanbackModule.as,  
 15 lns. 167, 668-77, 682-705, 1015-50; LeanbackV2Module.as, lns. 541-46; Dashboard.as, lns. 137,  
 16 143-48; Stations.as, lns. 15, 20-222. Each videoID is an “internal identifier to a video” that  
 17 “represents a video” and is used to access a video on a “Bandaid” server that is different than the  
 18 Lounge Server. Levai Dep. Tr. (May 4, 2022), 46:25-54:9; *see also* Levai Dep. Tr. (May 4, 2022),  
 19 52:1-6 (“I can say that the videoID is kind of like an internal identifier to a video, and it -- it  
 20 represents a video.”); Levai Decl., ¶3 (“Videos were played back on the television by retrieving  
 21 them from Bandaid content servers in Google’s Content Delivery Network.”). The Bandaid server  
 22 does not maintain any YTR playlists or queues.

171. In the YTR System, a remote control is used to create a playback queue that is stored  
 24 locally on the remote control. *See* Bhatta. Op. Report, ¶165 (“The YTR application allowed users  
 25 to create queues of videos.”). As shown below in images from Video #5 (GOOG-SONOSNDCA-  
 26 00071317) on the left and Video #6 (GOOG-SONOSNDCA-00071318) on the right, the “Queue”  
 27 on the remote control starts out “empty”:

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 back videos from its playback queue stored locally on the remote/mobile device. Levai Dep. Tr.  
 2 (January 6, 2023), 38:13-39:4, 40:15-41:9. The YTR System's standalone mode is discussed in  
 3 more detail below in connection with Dr. Bhattacharjee's reliance on Video #1 (GOOG-SONOS-  
 4 WDTX-INV-00015101).

5 174. To initiate the YTR System's remote-control mode, a remote sends a copy of its  
 6 entire local playback queue (*i.e.*, its locally-stored "list of media selected for playback") in a  
 7 SET\_PLAYLIST message (also referred to as a "setPlaylist" message) to the Lounge Server.  
 8 GOOG-SONOSWDTX-00040194; Levai Dep. Tr. (January 6, 2023), 39:13-43:12;  
 9 RemotePlayerController.java, lns. 92-102. The Lounge Server then sends the SET\_PLAYLIST  
 10 message to any and all of the connected Leanback Screens in the session, as well as to all other  
 11 remotes (if any) connected to the session. GOOG-SONOSWDTX-00040194 at 195; Levai Dep.  
 12 Tr. (January 6, 2023), 41:22-45:22; RealLougeSessionManager.java, lns. 307-86, 851-83, 1030-  
 13 42.

14 175. After receiving the SET\_PLAYLIST message from the Lounge Server, each  
 15 Leanback Screen saves a copy of the entire playlist into its local playback queue. Levai Dep. Tr.  
 16 (January 6, 2023), 41:10-15, 51:15-24, 58:16-19; LeanbackModule.as, lns. 167, 668-77, 682-705,  
 17 1015-50; LeanbackV2Module.as, lns. 541-46. Each Leanback Screen uses its local playback  
 18 queue, which now contains the entire playlist, to playback the media items sent from the remote  
 19 control. Levai Dep. Tr. (January 6, 2023), 52:5-14 59:15-60:12; LeanbackModule.as, lns. 167,  
 20 2144-82; LeanbackV2Module.as, lns. 541-46; VideoApplication.as, lns. 859-63.

21 176. Since the SET\_PLAYLIST message has no way of indicating which video in the  
 22 playlist should be played, the remote control will send a SET\_VIDEO message (also referred to as  
 23 a "setVideo" message) to the Lounge Server to indicate which video should be played on the  
 24 Leanback Screen(s). GOOG-SONOSWDTX-00040194; RealPlaystatePushService.java, lns. 31,  
 25 33-39, 116-35, 282-85, 264-80; RemotePlayerController.java, lns. 77-89, 92-102. The Lounge  
 26 Server, in turn, sends the SET\_VIDEO message to each Leanback Screen in the session. GOOG-  
 27 SONOSWDTX-00040194; RealLougeSessionManager.java, lns. 307-86, 851-83, 1030-42. To  
 28 change the currently playing video, the remote will need to send a new SET\_VIDEO message to

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 the Lounge Server, which will, in turn, send the new SET\_VIDEO message to each Leanback  
 2 Screen in the session. RealVideoPlayService.java, lns. 80-124, 249-67, 269-73;  
 3 RealPlaystatePushService.java, lns. 116-35, 282-85; RemotePlayerController.java, lns. 77-89.

4 177. As previously discussed, the connected remotes in the session have the ability to  
 5 modify or edit the playlist with updates. To do so, a remote will send an update message to the  
 6 Lounge Server, which will, in turn, send the updates for the playlist to the remotes in the session  
 7 and the Leanback Screens in the session via an UPDATE\_PLAYLIST message (also referred to as  
 8 a “updatePlaylist” message). GOOG-SONOSWDTX-00040194 at 194-195; Levai Dep. Tr.  
 9 (January 6, 2023), 51:6-21; RealPlaystatePushService.java, lns. 86-114, 305-53, 345-49;  
 10 RemotePlayerController.java, lns., 112-19; RealLoungeSessionManager.java, lns. 307-86, 851-83.

11 Once the updates have been processed by the Leanback Screens, the locally-stored playback queue  
 12 on each Leanback Screen will maintain the full updated playlist and each Leanback Screen will  
 13 continue to playback the media items from their locally-stored playback queue. Levai Dep. Tr.  
 14 (January 6, 2023), 52:5-14 59:15-60:12; LeanbackModule.as, lns. 682-705, 2144-82.

15 178. Accordingly, in the YTR System’s remote-control mode, the playback queue locally  
 16 stored on the Leanback Screen: (i) comprises the list of media items that is used for playback by  
 17 the Leanback Screen; (ii) contains the entire list of media items selected for playback; (iii) is not  
 18 being used merely to process a list of media items maintained elsewhere for playback; and (iv) is  
 19 the queue that “runs the show.”

20 179. In other words, if there is only one playback queue in the YTR System, it is the local  
 21 playback queue on the Leanback Screens. Dr. Bhattacharjee does not dispute this conclusion. *See*  
 22 Bhatta. Op. Report, ¶185.

23 **3. Dr. Bhattacharjee’s Mischaracterizations of the YTR System**

24 180. In his Opening Report, Dr. Bhattacharjee attempts to explain various aspects of the  
 25 YTR System. However, as explained below, I disagree with many of Dr. Bhattacharjee’s  
 26 characterizations.

27 181. For instance, as Dr. Bhattacharjee acknowledges in his Opening Report, the YTR  
 28 application served as a “remote control” for a larger screen. *See* Bhatta. Op. Report, ¶157; *see also*,

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

**a. Dr. Bhattacharjee's Lack of Evidence Regarding Party Mode Ever Working**

198. Other than source code, Dr. Bhattacharjee cites to no documents explaining how or if YTR party mode ever worked.<sup>17</sup> He also does not cite to any videos demonstrating how YTR party mode worked, or if it even worked at all. I am not aware of the existence of any such documents or videos. In fact, unlike the media coverage and user reviews that were posted on the Internet for the normal operation of YTR, I was unable to find any such coverage or reviews on YTR party mode. Given the lack of available information on YTR party mode, it is questionable whether YTR party mode was ever released for commercial use or whether it ever worked at all. At best, it appears that the YTR party mode was an unused feature that was merely present in the source code.

199. In my opinion, YTR party mode was merely an expansion of YTR's normal (or non-party) operating mode that was built on top of YTR's existing source code and functionality at the time. For instance, as explained above, YTR's non-party mode allowed multiple remote controls and multiple Leanback Screens to communicate with each other through a single Lounge Server. Similarly, YTR's party mode apparently did the same thing – it allowed multiple remote controls and multiple Leanback Screens to communicate with each other through a single Lounge Server. The only difference was that non-party mode involved remote controls and Leanback Screens logged into the same YouTube Account, while party mode purportedly permitted remote controls and Leanback Screens logged into different YouTube Accounts.

200. As set forth below, the relevant messages for setting up a playback queue in party mode are nearly identical to the messages used for non-party mode, except that the messages in party mode inform the Lounge Server that it needs to relay the playlist and updates to not just the remote controls and Leanback Screens associated with the host's YouTube account, but also to the remote controls and Leanback Screens associated with any guests' YouTube accounts in the same session. *See, e.g.*, Levai Dep. Tr. (January 6, 2023), 53:19-54:18 (describing the "set playlist"

<sup>17</sup> Dr. Bhattacharjee points to a document entitled “YouTube Remote Script.” Bhatta. Op. Report, ¶279. But that document does not show how or if YTR party mode ever worked. Rather, the script contains only a single sentence arguably related to party mode, which merely states that “Now I can invite my friends to join my screen, and we can collaboratively edit the Shared Queue.” *Id.*

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 messaging between the Lounge Server and Leanback Screen in party mode as “mostly the same  
 2 path as, yes, in non-Party Mode,” and confirming that, “based on the code,” in party mode “the  
 3 MDx server was mimicking the same messages that should be sent from the [] remote app when  
 4 [in] non-Party Mode”), 56:25-58:19 (confirming with an unqualified “yes” that “[i]n either Party  
 5 Mode or non-Party Mode, there is a set playlist message sent from the MDx Server down to the  
 6 Leanback screens” and that “[i]n both Party Mode and non-Party Mode, the Leanback screens  
 7 always get the entire list of videos for playback”). In other words, there is no substantive difference  
 8 between party mode and non-party mode except that, in party mode, the Lounge Server has to  
 9 coordinate with the guests’ devices in addition to the host’s devices.

10 201. I also disagree with Dr. Bhattacharjee’s opinion that YTR party mode relied on a  
 11 remote playback queue or a “cloud queue” for playback. *See* Bhatta. Op. Report, ¶186. To the  
 12 contrary, it is my opinion that YTR party mode used a local playback queue on the Leanback  
 13 Screens just like the normal, non-party mode of YTR.

14 202. Indeed, the copy of the playlist stored on the Lounge Server in party mode was not  
 15 used for playback. Rather, the Lounge Server used that copy of the playlist to merely “coordinate,”  
 16 *i.e.*, “make sure that every device had the same order of videos,” with the Leanback Screens and  
 17 the remote controls that were connected in the session. Levai Dep. Tr. (January 6, 2023), 56:2-  
 18 16.<sup>18</sup>

19 **b. Messages for Setting Up a Playback Queue on the Leanback  
 Screens in Party Mode**

20 203. As in non-party mode, it is important to note that there is no direct communication  
 21 between the remote controls (*e.g.*, mobile devices) and the Leanback Screens (*e.g.*, TVs) in the  
 22 YTR System’s party mode. Bhatta. Op. Report, ¶158. All communication between the remote  
 23 controls and the Leanback Screens must be facilitated through and by the Lounge Server in both  
 24 party and non-party modes. *Id.*

25 204. It is also important to note that in party mode only “videoIDs” are maintained in the

---

26 <sup>18</sup> Likewise, Dr. Bhattacharjee’s reference to comments in `RemoteQueueManager.java` (Bhatta.  
 27 Op. Report, ¶¶177, 300) is unconvincing because (i) that is a file for the YTR Application source  
 28 code (*i.e.*, a YTR remote control), (ii) as I explained in my Opening Report whether something is  
 “remote” of something else is context dependent, and (iii) here, the comment is referring to the  
 fact that the “remote queue” that the YTR application is managing is on the Screen(s).

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 playlist on the Lounge Server and the playback queues on the remote controls and Leanback  
2 Screens. Bhatta. Op. Report, ¶¶193-97; GOOG-SONOSWDTX-00040194; Levai Dep. Tr. (May  
3 4, 2022), 46:25-54:9; RealLoungSessionManager.java, lns. 63-1264 (197), 676-763, 679-88;  
4 RealPlaystatePushService.java, lns. 41-350 (57, 137-47) (bigScreenPlaylist); LeanbackModule.as,  
5 lns. 167, 668-77, 682-705, 1015-50; LeanbackV2Module.as, lns. 541-46; Dashboard.as, lns. 137,  
6 143-48; Stations.as, lns. 15, 20-222. Each videoIDs is an “internal identifier to a video” that  
7 “represents a video,” and is used to access a video on a “Bandaid” server that is different than the  
8 Lounge Server. Levai Dep. Tr. (May 4, 2022), 46:25-54:9; *see also* Levai Dep. Tr. (May 4, 2022),  
9 52:1-6 (“I can say that the videoID is kind of like an internal identifier to a video, and it -- it  
10 represents a video.”); Levai Decl., ¶3 (“Videos were played back on the television by retrieving  
11 them from Bandaid content servers in Google’s Content Delivery Network.”). The Bandaid server  
12 does not maintain any YTR playlists or queues.

13 205. The process for creating and populating the local playback queue on a remote  
14 control in party mode is the same as previously described with respect to non-party mode.

15 206. According to Dr. Bhattacharjee, “the party mode feature allowed the mobile device  
16 to play back a party queue stored in the cloud in standalone mode and ‘remote’ control mode as  
17 well.” Bhatta. Op. Report, ¶181. However, Dr. Bhattacharjee fails to explain or show how exactly  
18 the mobile devices operate in the YTR System’s standalone party mode. Nevertheless, I have seen  
19 no evidence that the remote controls/mobile devices in the party are doing anything but playing  
20 back the list of media items stored in their local playback queues—just like multiple remote controls  
21 would do in standalone non-party mode. Thus, I disagree with Dr. Bhattacharjee’s assertion that  
22 any mobile device in standalone party mode is playing back a queue that is “stored in the cloud.”

23 207. To initiate the YTR System’s remote-control party mode, a host remote control  
24 sends a copy of its entire local playback queue (*i.e.*, its locally-stored “list of media selected for  
25 playback”) in a SET\_PARTY\_PLAYLIST message to the Lounge Server. Bhatta. Op. Report,  
26 ¶189 (“If the YTR application is in party mode, this function constructs a  
27 SET\_PARTY\_PLAYLIST message (*id.*, 365) that includes the party queue (*id.*, lines 377-381),  
28 which is sent to the Lounge server.”); Levai Dep. Tr. (January 6, 2023), 52:15-

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 53:15; RealPartyModeManager.java, lns. 190-212; RealLoungSessionManager.java, lns. 307-86,  
 2 776-799, 1111-19, 1088-109, 1218-27. The host remote control may also change the value for the  
 3 playlist contain in its local playback queue from “QUEUE” to “PARTY\_QUEUE.”  
 4 RealPartyModeManager.java, lns. 190-212, 269-78. Otherwise, there is no substantive difference  
 5 between a playlist tagged as “QUEUE” versus “PARTY\_QUEUE.”

6 208. The Lounge Server saves a copy of the SET\_PARTY\_PLAYLIST message. Levai  
 7 Dep. Tr. (January 6, 2023), 52:15-53:15. The Lounge Server then sends a SET\_PLAYLIST  
 8 message with the playlist from the SET\_PARTY\_PLAYLIST message to any and all of the  
 9 connected Leanback Screens of the host and guests in the session, and sends a  
 10 PARTY\_PLAYLIST\_MODIFIED message to all other remote controls of the host and guests  
 11 connected to the session. Bhatta. Op. Report, ¶189 (“This sendPartyQueueToScreen function sends  
 12 a SET\_PLAYLIST [message] from the Lounge server to the connected Screens. The  
 13 SET\_PLAYLIST message sent from the Lounge server to the Screens contains the cloud hosted  
 14 party queue.”); Levai Dep. Tr. (January 6, 2023), 52:15-54:18, 57:13-58:6;  
 15 RealLoungSessionManager.java, lns. 676-763, 1111-19, 1218-27. The Lounge Server uses its  
 16 saved copy of the playlist from the SET\_PARTY\_PLAYLIST message to relay to any new  
 17 Leanback Screens and/or remote controls that join the existing party mode session a copy of the  
 18 playlist. Levai Dep. Tr. (January 6, 2023) at 52:15-53:15, 56:6-16;  
 19 RealLoungSessionManager.java, lns., 237-69, 1065-72, 1074-86, 1088-109.

20 209. The SET\_PLAYLIST message that each Leanback Screen receives from the Lounge  
 21 Server in party mode is the same as the SET\_PLAYLIST message that each Leanback Screen  
 22 receives from the Lounge Server in non-party mode. Bhatta. Op. Report, ¶193 (“As I discussed  
 23 above, transferring playback from a mobile device to a Screen causes the Lounge server to send a  
 24 SET\_PLAYLIST message and/or SET\_VIDEO message from the Lounge server to the Screens to  
 25 transfer playback. In particular, when not in party mode a SET\_PLAYLIST message is sent from  
 26 the Lounge Server to the Screens to send the queue and start playback of the queue. In party mode,  
 27 the Lounge Server sends a SET\_PLAYLIST message to the Screens to send the party queue and a  
 28 further SET\_VIDEO message is used to then start playback of the queue.”);

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 RealLoungeSessionManager.java, lns., 307-86, 851-83, 1030-42, 1088-109. After receiving this  
 2 SET\_PLAYLIST message from the Lounge Server, each Leanback Screen in the party mode  
 3 sessions saves a copy of the entire playlist into its local playback queue. Levai Dep. Tr. (January  
 4 6, 2023), 54:19-55:10, 58:16-9; LeanbackModule.as, lns. 167, 668-77, 682-705, 1015-50;  
 5 LeanbackV2Module.as, lns. 541-46. Just as in a non-party mode session, each Leanback Screen in  
 6 a party mode session uses its own local playback queue, which now contains the entire party  
 7 playlist, to playback the media items sent from the remote control(s). Bhatta. Op. Report, ¶189  
 8 (“The Screens receive the SET\_PLAYLIST message and perform additional processing and  
 9 messaging to play the videos in [their] queue . . .”); *id.*, ¶¶193-197; Levai Dep. Tr. (January 6,  
 10 2023), 54:19-55:10, 58:16-9; LeanbackModule.as, lns. 167, 2144-82; LeanbackV2Module.as, lns.  
 11 541-46; VideoApplication.as, lns. 859-63.

12 210. Mr. Levai’s deposition testimony confirms that a Leanback Screen in party mode  
 13 uses its own copy of the party playlist stored in its local queue for playback without any dependency  
 14 on the copy of the playlist on the Lounge Server:

15 Q Well, all the devices such as the Leanback screens and the YouTube remotes, they  
 16 also store and maintain a copy of that playlist; correct?

17 A Well, they need to do that so they can show it in [GUI] to the user.

18 Q And so they can play it back; right?

19 A Well, in remote mode, yes, the lounge screen plays back those videos.

20 Q Plays back those videos stored locally on the screen; right?

21 A Well, it gets a list of videos and which one to play and starts playing that video.  
 22 And then in order to know which video to play next when the current one ends, it  
 23 refers to that list.

24 Q The locally stored list on the Leanback screens; correct?

25 A Yes. I don’t believe it asked the MDx server for which video to play next when  
 26 the current ended. I believe it referred to that list that it caught before.

27 Levai Dep. Tr. (January 6, 2023), 59:15-60:12.

28 211. As long as the session is active, and absent any modifications or edits to the party  
 29 playlist on a remote control, each Leanback Screen in a party mode session will continue to  
 30 playback the media items in its locally-stored playback queue in the order set by that queue and

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 until that queue is empty. Levai Dep. Tr. (January 6, 2023), 55:19-25; LeanbackModule.as, lns. 167, 1435-50, 1308-64; VideoApplication.as, lns. 852-57.

212. Just like in non-party mode, the SET\_PARTY\_PLAYLIST and SET\_PLAYLIST messages in party mode have no way of indicating which video in the playlist should be played, and thus, a remote control will need to send a SET\_VIDEO message to the Lounge Server to indicate which video should be played on the Leanback Screen(s). Bhatta. Op. Report, ¶191 (“For instance, if a user presses the Connect/Reconnect button while in party mode, the YTR application will transfer playback to the Screens by sending a SET\_VIDEO message to the Lounge server.”); Levai Dep. Tr. (January 6, 2023), 54:11-18; RemotePlayerController.java, lns. 77-89. The Lounge Server, in turn, sends the SET\_VIDEO message to each Leanback Screen in the party mode session. Bhatta. Op. Report, ¶191 (“The SET\_VIDEO message is then relayed by the Lounge server to the Screens to transfer playback.”); RealLoungeSessionManager.java, lns. 307-86, 851-83, 1030-42. To change the currently playing video, the remote control will need to send a new SET\_VIDEO message to the Lounge Server, which will, in turn, send the new SET\_VIDEO message to each Leanback Screen in the party mode session. RemotePlayerController.java, lns. 77-89; RealLoungeSessionManager.java, lns. 307-86, 851-83, 1030-42.

213. As previously discussed with respect to a non-party mode session, the connected remote controls in a party mode session have the ability to modify or edit the playlist with updates. To do so, a remote control will send update messages to the Lounge Server, which will, in turn, send the updates for the playlist to the remote controls in the session via a PARTY\_PLAYLIST\_MODIFIED message and to the Leanback Screens in the session via an UPDATE\_PLAYLIST message. Levai Dep. Tr. (January 6, 2023), 55:11-18; SharedPlaylistContentService.java, lns. 30-122; RemoteQueueManager.java, lns. 16, 20-82; RealLoungeSessionManager.java, lns. 676-763, 689-760, 1121-24, 1218-27, 1229-38. Notably, the PARTY\_PLAYLIST\_MODIFIED message, as well as the UPDATE\_PLAYLIST message, “contains the entire shared party queue.” Bhatta. Op. Report, ¶175; RealLoungeSessionManager.java, lns. 1218-27, 1229-38.

214. The UPDATE\_PLAYLIST message that each Leanback Screen receives from the

**HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY**

1 Lounge Server in party mode is the same as the UPDATE\_PLAYLIST message that each Leanback  
 2 Screen receives from the Lounge Server in non-party mode. RemotePlayerController.java lns. 112-19; RealLoungeSessionManager.java, lns. 1229-38. Once the updates have been processed by the  
 3 Leanback Screens, the locally-stored playback queue on each Leanback Screen in the party mode  
 4 session will maintain the full updated playlist and each Leanback Screen in the party mode session  
 5 will continue to playback the media items from their locally-stored playback queue. Levai Dep.  
 6 Tr. (January 6, 2023), 54:19-55:10, 58:16-9, 59:15-60:12; LeanbackModule.as, lns. 167, 682-705, 1435-50, 1308-64; VideoApplications.as, lns. 852-57.

9       215. Accordingly, in the YTR System's remote-control party mode, the playback queue  
 10 locally stored on the Leanback Screen: (i) comprises the list of media items that is used for playback  
 11 by the Leanback Screen; (ii) contains the entire list of media items selected for playback; (iii) is not  
 12 being used merely to process a list of media items maintained elsewhere for playback; and (iv) is  
 13 the queue that "runs the show."

14       216. In other words, as with non-party mode, if there is only one playback queue in the  
 15 YTR System's party mode, it is the local playback queue on the Leanback Screens.

16       **B. U.S. Patent No. 9,490,998**

17       217. Dr. Bhattacharjee relies on U.S. Patent No. 9,490,998 (the "'998 Patent") to support  
 18 his YTR obviousness theories, namely, that it would have been obvious to add a "device-picker"  
 19 to the YTR System. I disagree with Dr. Bhattacharjee that the '998 Patent discloses a device-picker  
 20 as explained in my Rebuttal Report for the Court's Patent Showdown Procedure, which is  
 21 incorporated herein by reference. I also disagree with Dr. Bhattacharjee that it would have been  
 22 obvious to combine the YTR System with the '998 Patent to render the Asserted Claims of the '033  
 23 Patent obvious. In fact, the YTR System's party mode is completely incompatible and inoperable  
 24 with a device-picker.

25       218. The '998 Patent is entitled "Network-Based Remote Control" and issued from an  
 26 application filed on March 7, 2011, which claims priority to November 8, 2010. The '998 Patent  
 27 lists Google as the assignee. One of the named inventors of the '998 Patent is Ms. Bobohalma,  
 28 who stated, in her declaration, that "[t]his patent discloses *some* of the work that I did on the